Tri_Group_inflation_comparison

February 13, 2025

1 Setup and Data

[1]:
```
from inflation_analysis import calculate_price_indexes, tri_grouping,␣
 ↪output_data, output_obs_table, price_index_over_time,␣
 ↪top_abs_weight_differences, top_price_index_contributors,␣
 ↪tri_grouping_extended
```

[2]:
```
# Parameters
start_year = 2021
end_year = 2022
data_folder="/Users/roykisluk/Downloads/Consumer_Expenditure_Survey/"
top_n = 10
base_year = start_year
comparison_year = end_year

# Grouping
if start_year >= 2014: # After 2014, it is possible to identify secular,␣
 ↪conservative, and religious groups
    demo, income, ses, total_mmb = tri_grouping_extended(start_year, end_year,␣
 ↪cex_data_folder = data_folder)
else:
    demo, income, ses, total_mmb = tri_grouping(start_year, end_year,␣
 ↪cex_data_folder = data_folder)
```

[3]:
```
# Prepare data: calculate price indexes for each group, secondary and primary␣
 ↪categories, and total
demo_analysis, demo_mmb = output_data(demo, start_year, end_year, base_year,␣
 ↪top_n, data_folder)
income_analysis, income_mmb = output_data(income, start_year, end_year,␣
 ↪base_year, top_n, data_folder)
ses_analysis, ses_mmb = output_data(ses, start_year, end_year, base_year,␣
 ↪top_n, data_folder)

# General population
print("Calculating price indexes for general population...")
```

```
gen_pop_df, gen_pop_secondary_df, gen_pop_primary_df,␣
 ↪gen_pop_yearly_price_index = calculate_price_indexes(start_year, end_year,␣
 ↪base_year, cex_data_folder=data_folder, verbose=False)
gen_pop = {
    'combined_secondary_df': gen_pop_secondary_df,
    'combined_primary_df': gen_pop_primary_df,
    'yearly_price_index': gen_pop_yearly_price_index
}
print("Done.")
```

```
Group 1/7 (Secular) started.

Loading price data: 100%|      | 2/2 [00:00<00:00,  2.16it/s]
Calculating price indexes: 100%|      | 2/2 [00:01<00:00,  1.02it/s]

Group 1/7 (Secular) successfully computed.
Group 2/7 (Conservative) started.

Loading price data: 100%|      | 2/2 [00:00<00:00,  2.96it/s]
Calculating price indexes: 100%|      | 2/2 [00:01<00:00,  1.26it/s]

Group 2/7 (Conservative) successfully computed.
Group 3/7 (Religious) started.

Loading price data: 100%|      | 2/2 [00:00<00:00,  3.16it/s]
Calculating price indexes: 100%|      | 2/2 [00:01<00:00,  1.59it/s]

Group 3/7 (Religious) successfully computed.
Group 4/7 (Haredi) started.

Loading price data: 100%|      | 2/2 [00:00<00:00,  3.20it/s]
Calculating price indexes: 100%|      | 2/2 [00:01<00:00,  1.90it/s]

Group 4/7 (Haredi) successfully computed.
Group 5/7 (Arabs) started.

Loading price data: 100%|      | 2/2 [00:00<00:00,  3.19it/s]
Calculating price indexes: 100%|      | 2/2 [00:01<00:00,  1.76it/s]

Group 5/7 (Arabs) successfully computed.
Group 6/7 (Young) started.

Loading price data: 100%|      | 2/2 [00:00<00:00,  3.15it/s]
Calculating price indexes: 100%|      | 2/2 [00:01<00:00,  1.75it/s]

Group 6/7 (Young) successfully computed.
Group 7/7 (Old) started.

Loading price data: 100%|      | 2/2 [00:00<00:00,  3.18it/s]
Calculating price indexes: 100%|      | 2/2 [00:01<00:00,  1.50it/s]

Group 7/7 (Old) successfully computed.
Group 1/10 (1) started.
```

```
Loading price data: 100%|     | 2/2 [00:00<00:00,  3.18it/s]
Calculating price indexes: 100%|     | 2/2 [00:00<00:00,  2.28it/s]

Group 1/10 (1) successfully computed.
Group 2/10 (2) started.

Loading price data: 100%|     | 2/2 [00:00<00:00,  3.19it/s]
Calculating price indexes: 100%|     | 2/2 [00:00<00:00,  2.19it/s]

Group 2/10 (2) successfully computed.
Group 3/10 (3) started.

Loading price data: 100%|     | 2/2 [00:00<00:00,  3.15it/s]
Calculating price indexes: 100%|     | 2/2 [00:00<00:00,  2.05it/s]

Group 3/10 (3) successfully computed.
Group 4/10 (4) started.

Loading price data: 100%|     | 2/2 [00:00<00:00,  3.20it/s]
Calculating price indexes: 100%|     | 2/2 [00:00<00:00,  2.10it/s]

Group 4/10 (4) successfully computed.
Group 5/10 (5) started.

Loading price data: 100%|     | 2/2 [00:00<00:00,  3.19it/s]
Calculating price indexes: 100%|     | 2/2 [00:00<00:00,  2.11it/s]

Group 5/10 (5) successfully computed.
Group 6/10 (6) started.

Loading price data: 100%|     | 2/2 [00:00<00:00,  3.19it/s]
Calculating price indexes: 100%|     | 2/2 [00:00<00:00,  2.05it/s]

Group 6/10 (6) successfully computed.
Group 7/10 (7) started.

Loading price data: 100%|     | 2/2 [00:00<00:00,  3.20it/s]
Calculating price indexes: 100%|     | 2/2 [00:00<00:00,  2.05it/s]

Group 7/10 (7) successfully computed.
Group 8/10 (8) started.

Loading price data: 100%|     | 2/2 [00:00<00:00,  3.20it/s]
Calculating price indexes: 100%|     | 2/2 [00:01<00:00,  1.98it/s]

Group 8/10 (8) successfully computed.
Group 9/10 (9) started.

Loading price data: 100%|     | 2/2 [00:00<00:00,  3.18it/s]
Calculating price indexes: 100%|     | 2/2 [00:01<00:00,  1.89it/s]

Group 9/10 (9) successfully computed.
Group 10/10 (10) started.

Loading price data: 100%|     | 2/2 [00:00<00:00,  3.19it/s]
Calculating price indexes: 100%|     | 2/2 [00:01<00:00,  1.84it/s]
```

```
Group 10/10 (10) successfully computed.
Group 1/5 (1) started.

Loading price data: 100%|     | 2/2 [00:00<00:00,  3.20it/s]
Calculating price indexes: 100%|     | 2/2 [00:01<00:00,  1.56it/s]

Group 1/5 (1) successfully computed.
Group 2/5 (2) started.

Loading price data: 100%|     | 2/2 [00:00<00:00,  3.19it/s]
Calculating price indexes: 100%|     | 2/2 [00:01<00:00,  1.84it/s]

Group 2/5 (2) successfully computed.
Group 3/5 (3) started.

Loading price data: 100%|     | 2/2 [00:00<00:00,  3.19it/s]
Calculating price indexes: 100%|     | 2/2 [00:01<00:00,  1.48it/s]

Group 3/5 (3) successfully computed.
Group 4/5 (4) started.

Loading price data: 100%|     | 2/2 [00:00<00:00,  3.18it/s]
Calculating price indexes: 100%|     | 2/2 [00:01<00:00,  1.21it/s]

Group 4/5 (4) successfully computed.
Group 5/5 (5) started.

Loading price data: 100%|     | 2/2 [00:00<00:00,  3.19it/s]
Calculating price indexes: 100%|     | 2/2 [00:00<00:00,  2.99it/s]

Group 5/5 (5) successfully computed.
Calculating price indexes for general population…

Loading price data: 100%|     | 2/2 [00:00<00:00,  3.19it/s]
Calculating price indexes: 100%|     | 2/2 [00:02<00:00,  1.48s/it]

Done.
```

2 Output

2.1 Tables

[4]:
```
# Observations tables
df=output_obs_table(start_year, end_year, demo_mmb)
print(df.to_latex(index=False))
```

```
+--------------+---------------+---------------+
|              | 2021          | 2022          |
|--------------+---------------+---------------|
| Secular      | 2690 (44.65%) | 2294 (41.99%) |
| Conservative | 1577 (26.18%) | 1602 (29.32%) |
| Religious    | 1035 (17.18%) | 808 (14.79%)  |
```

```
| Haredi        | 551 (9.15%)   | 595 (10.89%)  |
| Arabs         | 951 (15.79%)  | 727 (13.31%)  |
| Young         | 877 (14.56%)  | 820 (15.01%)  |
| Old           | 1779 (29.53%) | 1663 (30.44%) |
| Total         | 6024 (100.0%) | 5463 (100.0%) |
+---------------+---------------+---------------+
\begin{tabular}{ll}
\toprule
2021 & 2022 \\
\midrule
2690 (44.65%) & 2294 (41.99%) \\
1577 (26.18%) & 1602 (29.32%) \\
1035 (17.18%) & 808 (14.79%) \\
551 (9.15%) & 595 (10.89%) \\
951 (15.79%) & 727 (13.31%) \\
877 (14.56%) & 820 (15.01%) \\
1779 (29.53%) & 1663 (30.44%) \\
6024 (100.0%) & 5463 (100.0%) \\
\bottomrule
\end{tabular}
```

[ ]:
```
output_obs_table(start_year, end_year, income_mmb)
```

[ ]:
```
output_obs_table(start_year, end_year, ses_mmb)
```

2.2 Plots

2.2.1 Yearly Price Index Comparison Between Groups

[ ]:
```
price_index_over_time(demo_analysis)
```

[ ]:
```
price_index_over_time(income_analysis)
```

[ ]:
```
price_index_over_time(ses_analysis)
```

2.2.2 Top Weight Differences

[ ]:
```
# Define control group
weights_comparison_control =␣
 ↪gen_pop['combined_secondary_df'][gen_pop['combined_secondary_df']['Year'] ==␣
 ↪comparison_year]
```

[ ]:
```
# Top weight differences - demograhic groups
demo_comparison_groups = {}
for group in demo_analysis:
```

```
    demo_comparison_groups[group] =␣
 ↪demo_analysis[group]['combined_secondary_df'][demo_analysis[group]['combined_secondary_df']
 ↪== comparison_year]
top_abs_weight_differences(demo_comparison_groups, weights_comparison_control,␣
 ↪top_n)
```

[ ]:
```
# Top weight differences - income groups
income_comparison_groups = {}
for group in income_analysis:
    income_comparison_groups[group] =␣
 ↪income_analysis[group]['combined_secondary_df'][income_analysis[group]['combined_secondary_
 ↪== comparison_year]
top_abs_weight_differences(income_comparison_groups,␣
 ↪weights_comparison_control, top_n)
```

[ ]:
```
# Top weight differences - SES groups
ses_comparison_groups = {}
for group in ses_analysis:
    ses_comparison_groups[group] =␣
 ↪ses_analysis[group]['combined_secondary_df'][ses_analysis[group]['combined_secondary_df']['
 ↪== comparison_year]
top_abs_weight_differences(ses_comparison_groups, weights_comparison_control,␣
 ↪top_n)
```

2.2.3 Top Contributors to CPI Change

[ ]:
```
# Top contributors - demographic groups
demo_yearly_price_indexes = {}
for group in demo_analysis:
    demo_yearly_price_indexes[group] =␣
 ↪demo_analysis[group]['yearly_price_index'][comparison_year]
top_price_index_contributors(demo_comparison_groups, demo_yearly_price_indexes,␣
 ↪top_n)
```

[ ]:
```
# Top contributors - income groups
income_yearly_price_indexes = {}
for group in income_analysis:
    income_yearly_price_indexes[group] =␣
 ↪income_analysis[group]['yearly_price_index'][comparison_year]
top_price_index_contributors(income_comparison_groups,␣
 ↪income_yearly_price_indexes, top_n)
```

[ ]:
```
# Top contributors - SES groups
ses_yearly_price_indexes = {}
for group in ses_analysis:
    ses_yearly_price_indexes[group] =␣
 ↪ses_analysis[group]['yearly_price_index'][comparison_year]
```

```
top_price_index_contributors(ses_comparison_groups, ses_yearly_price_indexes,␣
↪top_n)
```

[ ]:
```
!jupyter nbconvert --to pdf Tri_Group_inflation_comparison.ipynb
```